Information Sciences & Technology
Ames Research Center

Discovery of Anomalies on Liquid Propulsion Systems: Update on Status

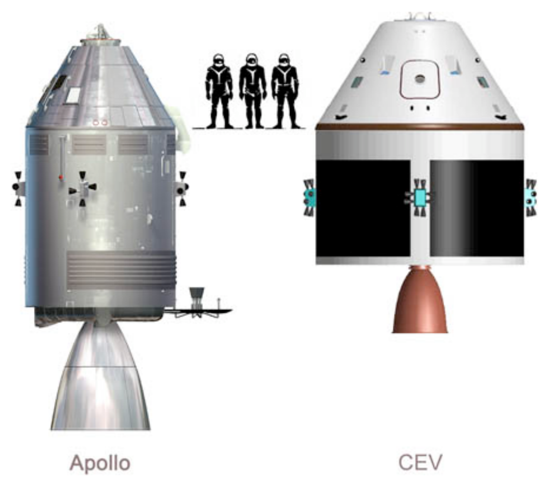
NASA Ames Team:
Ashok Srivastava, Ph.D.§
Mark Schwabacher, Ph.D.
Nikunj Oza, Ph.D.
Rodney Martin, Ph.D.
Richard Watson
Bryan Matthews

§Project Lead and POC: Send comments to ashok@email.arc.nasa.gov

Background: Target and Current Testbeds

- SSME – Have test stand data from Rocketdyne to design algorithms that will aid in the early detection of impending failures during operation. Serves as a testbed for CEV/CLV.

- CEV/CLV – Methods implemented on SSME data will be improved, extended, and used for future platforms.



CLV

Near-Term Mission Objectives for ARC

- Demonstrate ability to detect one critical failure mode within remediation window; this 2 sec window of time is dependent upon the failure mode selected and the response available to the selected propulsion system. (WBS 2.2.1.1, Due Jan' 07)

- Benchmark algorithm computational impacts and performance, implementable in real-time operation: Baseline performance metrics of the current failure detection algorithm.
  (WBS 2.2.1.2, Due Apr' 07)

- HM Technology Infusion Demonstration: Proposed Plan, Demonstration Report
  (WBS 2.4.4, Due FY07 Q3)

- Sensor Failure Robustness Demonstration: Demonstration for failure detection algorithm performance with individual sensor failure present. What is the impact on the benchmarked performance metrics when sensor failures are encountered ? The sensor failures may be single or multiple failure to occur during the simulation. Sensor failures may occur simultaneously or within demonstration window.
  (WBS 2.2.1.3, Due Aug' 07)

Team Approach

- How do we achieve these objectives/provide these deliverables ?
  - Need An Algorithmic Suite – Try multiple approaches/methods in attacking the problem
  - Need More Data - Further examples of anomalies
  - Need Performance Measures – How well do the designed algorithms detect anomalies ?
- Use the entire continuum from theoretical/simulated anomalies to real/actual anomalies (focus areas are highlighted in blue)

1. Synthetic data: We generate synthetic data that we believe to have similar properties to the real failure data. We could also call this a "low-fi simulator."
2. Hi-fi simulators: We use data from a high-fidelity simulator that was developed by rocket engine experts. Such simulators exist at Rocketdyne, MSFC, and GRC. An advantage over historical data is that faults that have never occurred in reality can be simulated.
3. Historical data: Since the CLV doesn't exist yet, we use historical SSME flight or test-stand data instead.
4. Actual test stand data: Obtain data with failures introduced during operation.

Algorithmic Approaches

- Data-Driven Failure/Fault Detection Algorithmic Development Using the Following Methods
  - Unsupervised Learning: These algorithms take only nominal data as input, and learn a model of the nominal data. When future data fails to match the learned model, they signal an anomaly.
  - Supervised Learning: These algorithms take as input labeled examples of nominal data and failure data, and learn a model that distinguishes between the two.
  - Semi-Supervised Learning: These algorithms take as input labeled nominal data, labeled failure data, and unlabeled data, and seek to take advantage of all three in order to build a model that can distinguish between nominal data and failure data.

Status of Data Processing Toolkits (Software Infrastructure) Supporting Analysis and Investigation

ISHM Discovery Work Bench

- ISHM-DMB (Integrated Systems Health Management – Discovery Work Bench) is a generalized framework for exploring and analyzing both data and data mining algorithms. It provides a streamlined, consistent and modular way to:
  - Import and reformat data
  - Write, parameterize and apply new and existing algorithms to this data
  - Visualize both datasets and algorithm functionality
  - Benchmark performance of algorithms
  - Interactively explore data and algorithms

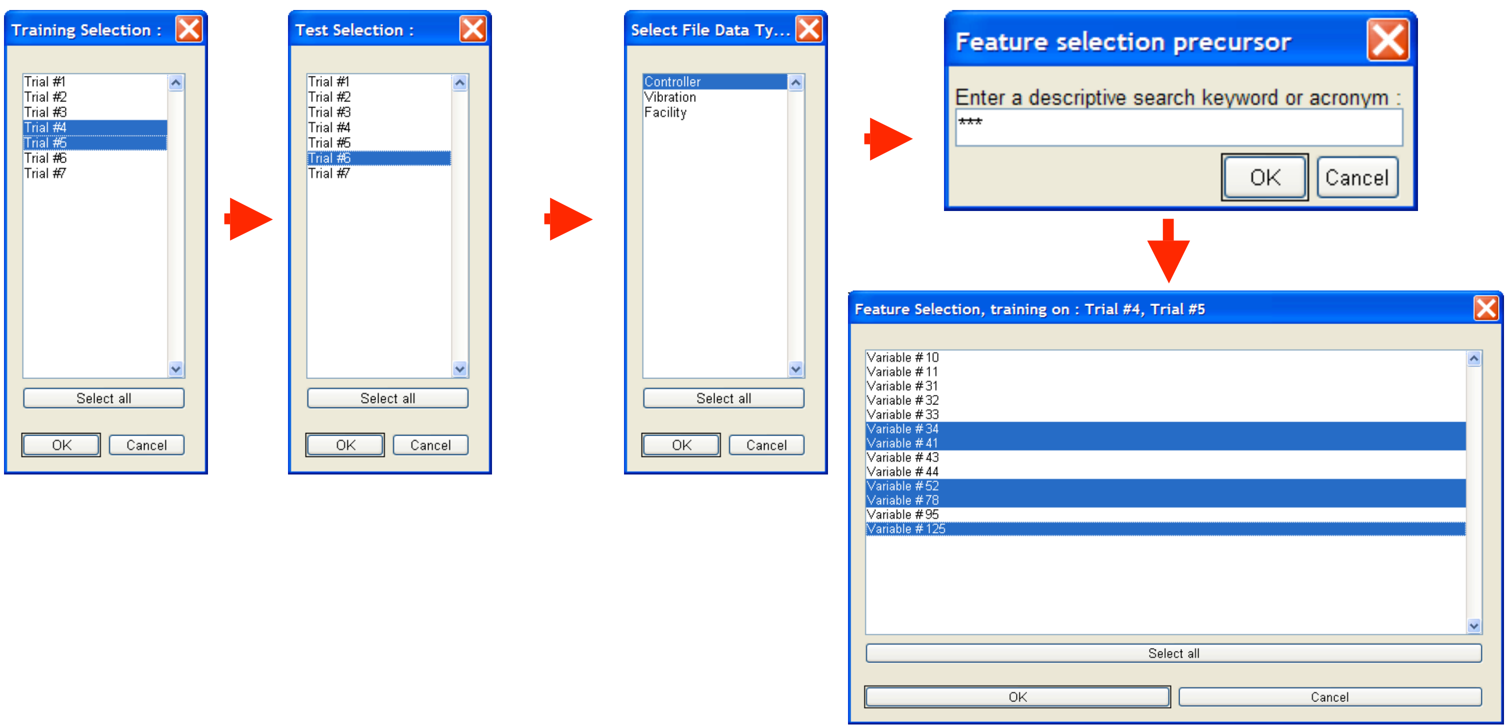
- We are using it prognostically to detect and predict anomalies, mode changes and artifacts in SSME data sets and to build algorithms that can automate that process.

- A suite of supervised learning algorithms have already been parameterized and incorporated into the framework
  - Gaussian processes
  - Linear prediction
  - Quadratic prediction
  - Bagged neural networks

- Additionally, several more algorithms are currently being incorporated into the system. These include:
  - A Semi-supervised algorithm, based upon SVM (Support Vector Machines)
  - Unsupervised methods:
    - GMM (Gaussian Mixture Models)
    - HMM (Hidden Markov Models)
    - KF (Kalman Filters)

MATLAB SSME Data Loader for Unsupervised Learning

- For Unsupervised GMM, HMM, KF Methods, MATLAB- Based SSME Data Tools Already Exist (to be hooked to ISHM-DMWB)
  - Loads in data from available Rocketdyne tests, processes the data prior to training



Applying New Algorithmic Approaches (Semi-Supervised, Unsupervised)

Semi-Supervised Learning Status

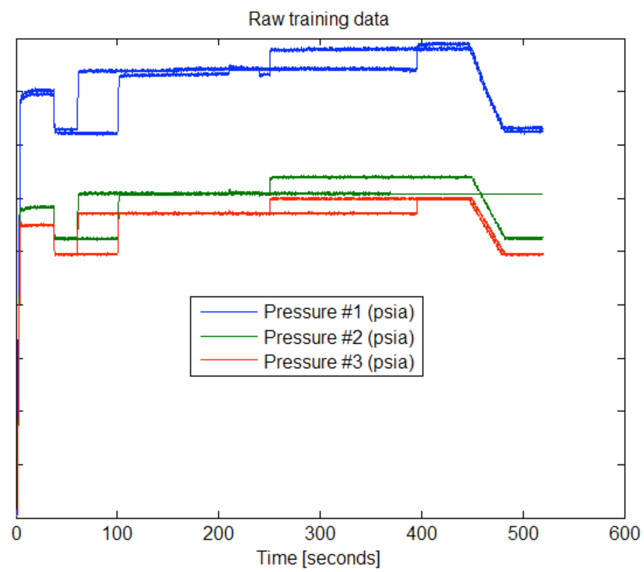
- The algorithm takes both expert labeled and unlabeled observations and creates a model, which can be used to predict unseen data.
- This semi-supervised learning algorithm can be useful in a data set such as SSME, which may have a very small number of expert labeled observations.
- The algorithm has been integrated into the ISHM-DWB for batch testing of multiple data sets.

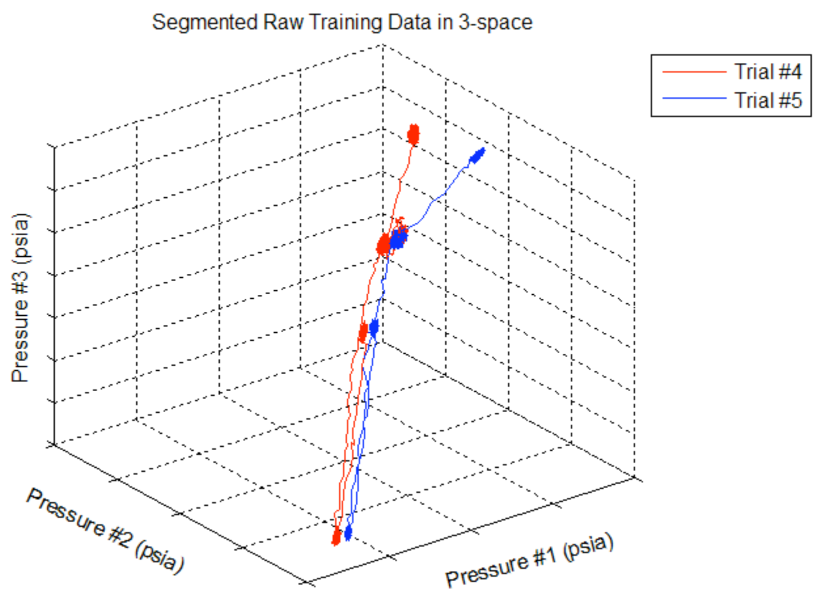
Unsupervised Learning Status

- New Methods
  - GMM (Gaussian Mixture Model)…In Progress
  - HMM (Hidden Markov Model)…Not started
  - KF (Kalman Filtering)…Not started
- These methods characterize nominal behavior, so we can detect off-nominal behavior
- They provide the means by which to measure performance of systems for a given design criteria
  - Minimum Allowable Probability of False Alarms
  - Threshold-based methods

Unsupervised Learning Status

- New Methods – Preliminary Results for GMM
- Train on 3 Sensor Data Values Relevant to a particular subsystem
- In "Raw training data" graph, each of 3 sensors is represented by 2 independent, superimposed runs
- In "Segmented Raw Training Data in 3-space," notice how well (4) clusters form for GMM training



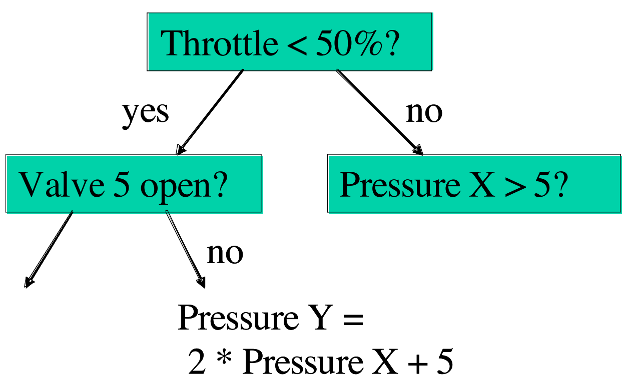


Results for Remaining Algorithmic Approaches
(Unsupervised, Virtual Sensors: Pseudo-Supervised)

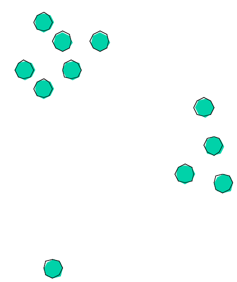
Unsupervised Learning Status

- Previous Methods
  - M. Schwabacher. Machine Learning for Rocket Propulsion Health Monitoring. SAE World Aerospace Congress, 2005.
  - Orca (Bay & Schwabacher, 2003)
    - uses a nearest-neighbor approach
    - uses a novel pruning rule to run in nearly linear time
  - GritBot
    - commercial product from RuleQuest Research
    - uses a decision-tree approach

GritBot: Decision trees

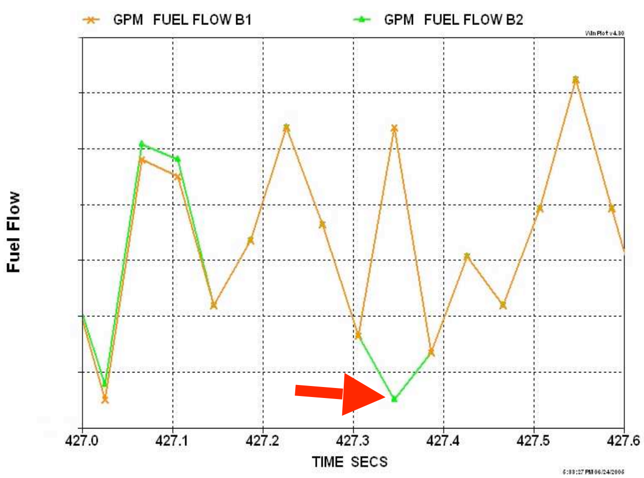


Orca: Nearest Neighbors

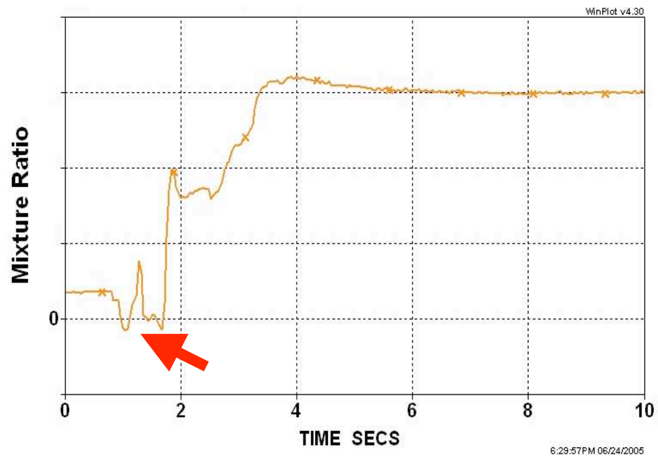


Unsupervised Learning Status

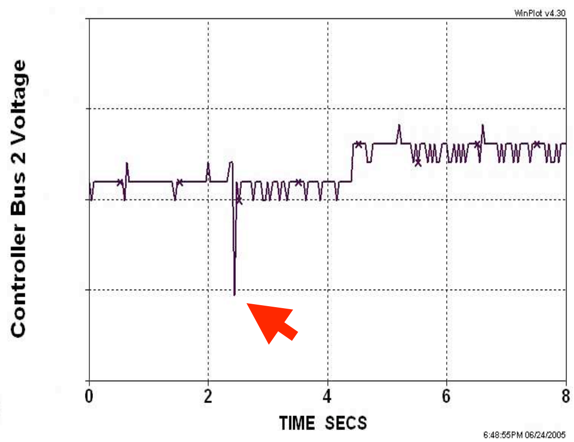
- Previous Results
  - M. Schwabacher. Machine Learning for Rocket Propulsion Health Monitoring. SAE World Aerospace Congress, 2005.
  - GritBot/Orca: Used for detection of Artifacts and Mode Changes



GritBot: Unexplained, but considered harmless



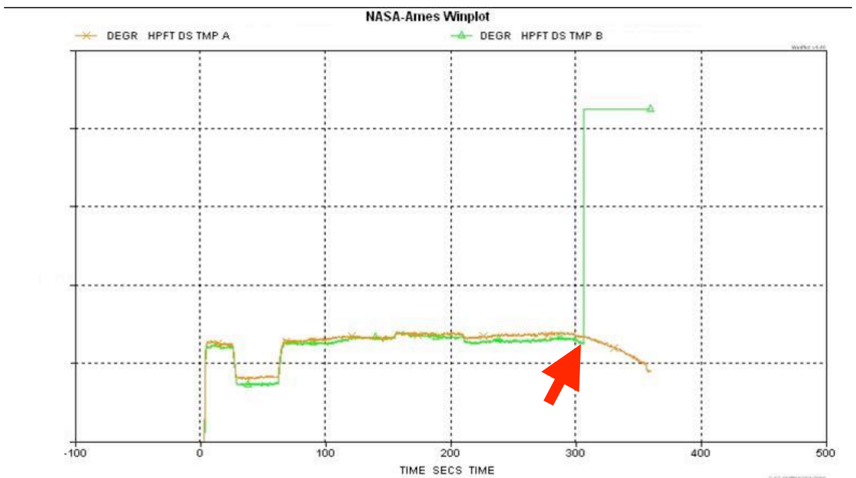
GritBot: Known artifact of the way in which mixture ratio is calculated (no oxidizer flow meter).



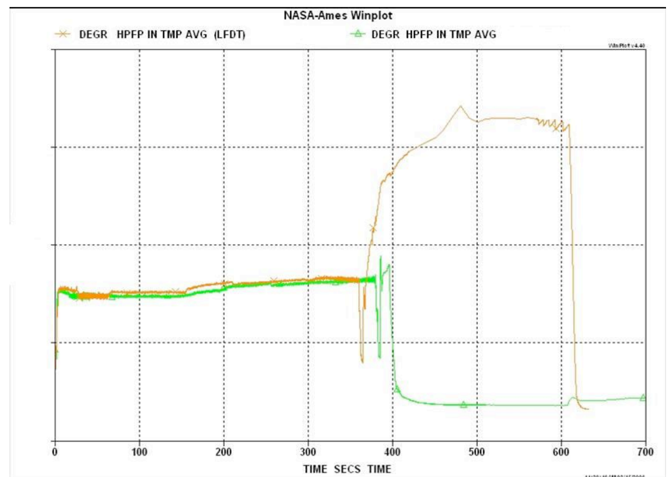
Orca: Caused by main igniters firing.

Unsupervised Learning Status

- Deviations from normal system or sensor behavior



GritBot: Temperature Sensor Failure

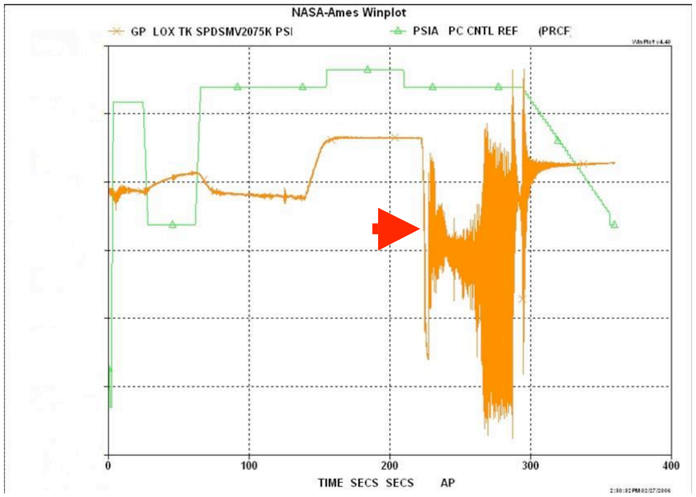


Orca: Unexplained difference in temperature profile§

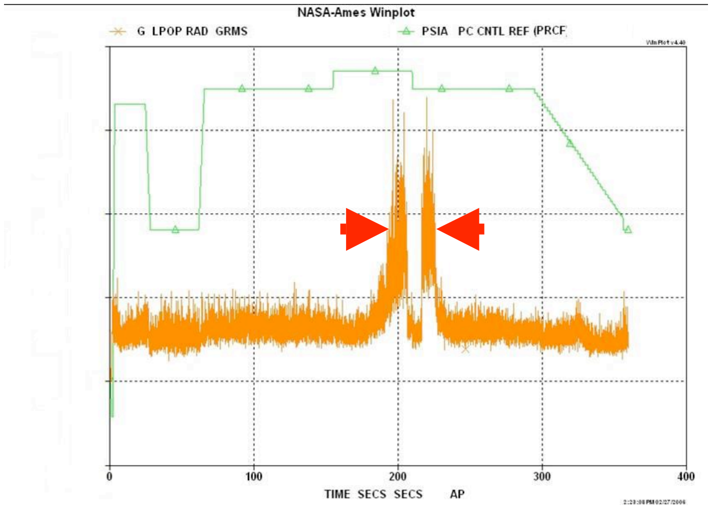
- Redundant sensors shown in left graph
- §Remains unvalidated by domain experts

Unsupervised Learning Status

- Detection of Sensor Failures



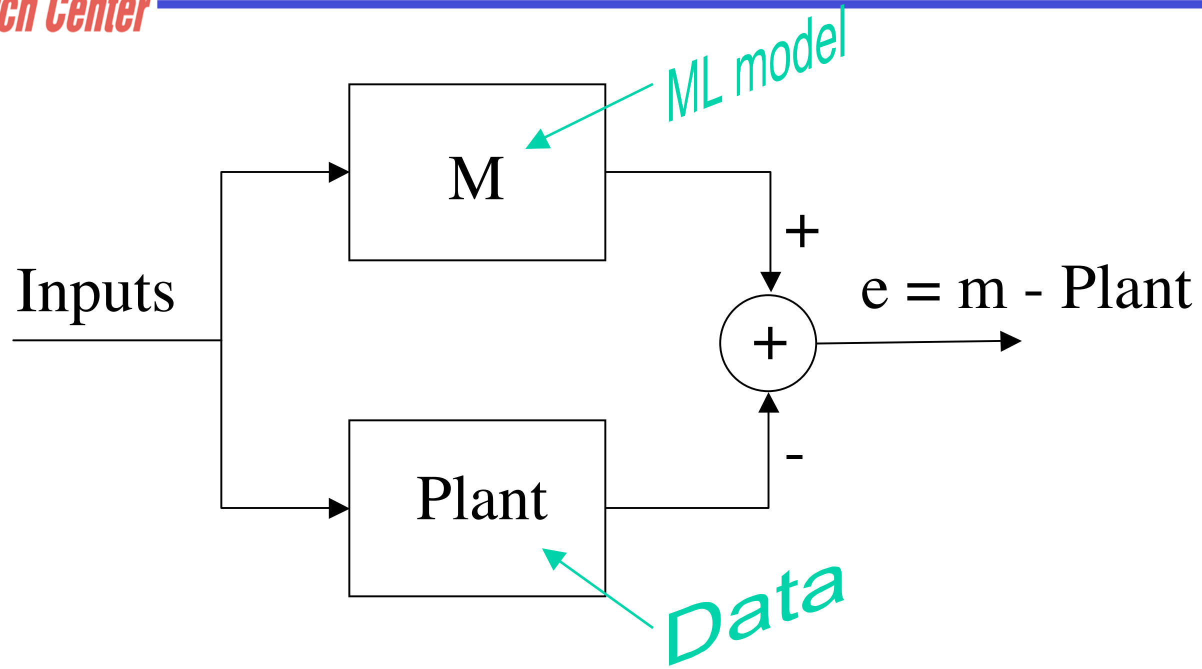
Orca: Unexplained aberration in pressure sensor§



Orca: Unexplained aberration in vibration sensor§

- §Remains unvalidated by domain experts

Virtual Sensors



- Hypothesis: Changes in the operating mode of a system (e.g., from normal to abnormal) will manifest themselves as changes in the relationships between measurements.
- We monitor difference between real measurement given in the data and estimate returned by machine learning model (e.g., neural network).

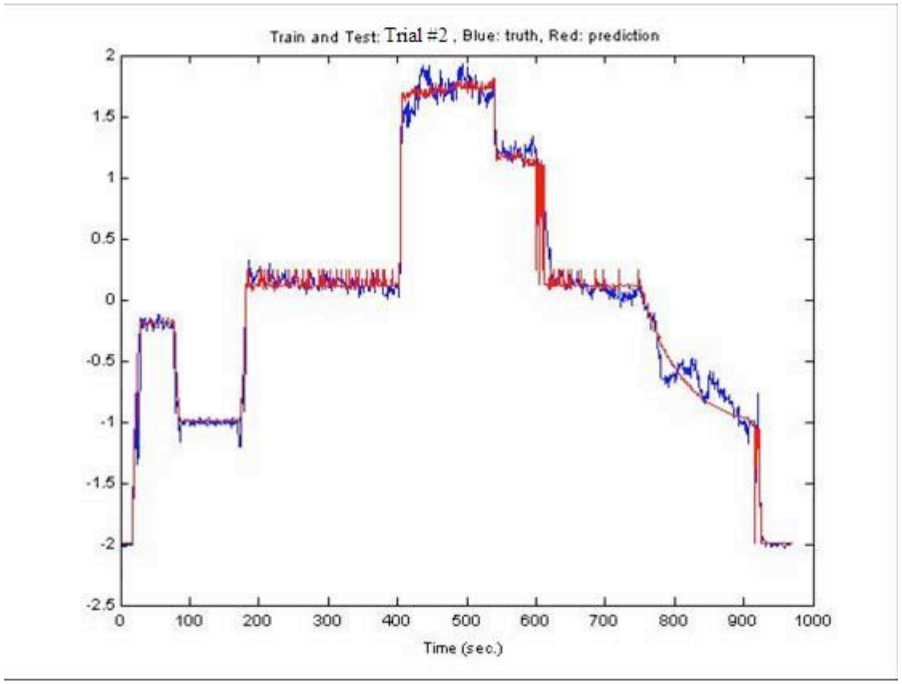
More Possible Uses of Virtual Sensors

- Use estimated measurements when real measurements ("Plant") are missing or corrupted.
- Estimate measurements for past systems where such measurements were not available.
  - Train model to predict a new measurement using data from later tests.
  - Use model to generate estimates of that new measurement on earlier tests.
  - Assumes that system did not change significantly across tests.

Preliminary Results

- Two tests used:
  - Trial # 1: Known not to have knife-edge seal crack.
  - Trial # 2: Known to have knife-edge seal crack.
- Response vibration data, each column z-scored (subtract mean and divide by standard deviation), resulting columns added.
- Predictor controller data: throttle, accelerometers (2 locations). Only points corresponding to times available in vibration file were selected. Time range: 0-400 seconds, increment 0.4 seconds (downsampled from 0.02 seconds in Trial # 1, 0.04 seconds in Trial #2).
- Model used: MultiLayer Perceptron (MLP), 3 hidden units, 500 epochs (iterations through training set).

Preliminary Results

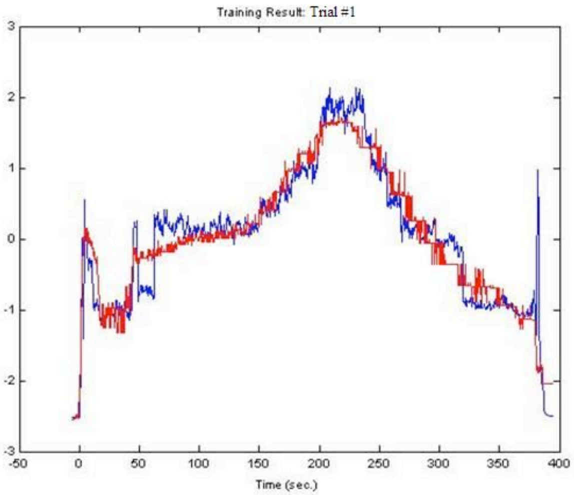


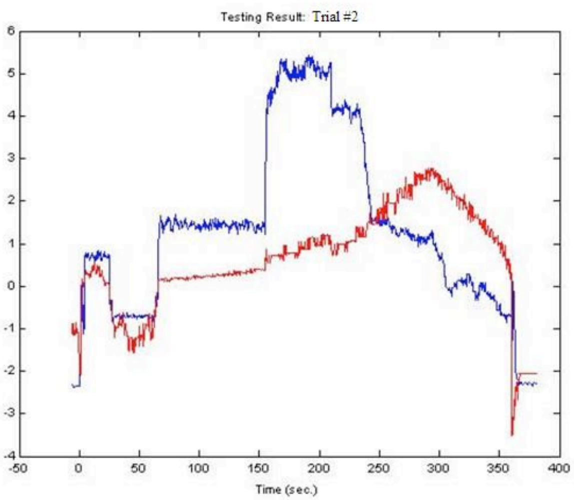
Trained and tested on Trial #2 to see if modeling is possible.

Blue: truth, Red: prediction

87% of variance explained.

Preliminary Results





- Trained on Trial # 1 (results in top graph), tested on Trial #2 (results in bottom graph).
- Crack thought to occur in Trial #2 in 100-150 second range.
- Significant change in residual starting at 65 seconds. Bigger change at 150 seconds.
- Real reason TBD.

Conclusions

- All results shown are still preliminary
- Still further validation to perform
- We have some promising new areas of development that are potentially fruitful for future investigation in prognostics and setting performance measures